

# A PARALLEL PREPROCESSING FOR THE OPTIMAL ASSIGNMENT PROBLEM BASED ON DIAGONAL SCALING

MEISAM SHARIFY\*, STÉPHANE GAUBERT<sup>†</sup>, AND LAURA GRIGORI<sup>‡</sup>

**Abstract.** We present a preprocessing method, which is suitable for parallel computation, to solve large optimal assignment problems. We think of the optimal assignment problem as a limit of a deformation of an entropy maximization problem. We show that the matrix maximizing the entropy converges, as the deformation parameter goes to infinity, to a matrix whose nonzero entries are precisely the ones belonging to optimal assignments. For every value of the deformation parameter, the matrix of maximal entropy can be computed by Sinkhorn iteration. This leads to a parallel preprocessing for the optimal assignment problem, which allows to delete entries that do not belong to optimal assignments, so that the reduced problem becomes executable on a sequential machine.

**Key words.** large scale optimal assignment problem, entropy maximization, matrix diagonal scaling, parallel computing, Sinkhorn iteration, Newton method

**AMS subject classifications.** 90C27, 54C70, 65Y05, 90C06, 49M15

**1. Introduction.** One of the most classical problems in combinatorial optimization is the optimal assignment problem. Several applications of this problem arise in different fields of applied sciences such as bioinformatics for protein structure alignment problem [Hol93, LCL04], VLSI design [HCLH90], image processing and computer vision [CWC<sup>+</sup>96], and the pivoting problem in the solution of large linear systems of equations [ON96, DK00, LD03]. Thus, this problem has received considerable attention and several algorithms have been proposed to solve it.

The first polynomial time algorithm to solve this problem was proposed by H. W. Kuhn in 1955 [Kuh55]. It works in  $O(n^4)$  time, which was improved to  $O(n^3)$  by Edmonds and Karp [EK70] (see also [DK69]). In the sparse case, Fredman and Tarjan [FT87] proposed an improved algorithm which uses Fibonacci heaps for the shortest paths computations. It runs in  $O(n(m + n \log n))$  time. Several other algorithms have also been developed. We refer the interested reader to the recent book of Burkard et al. [BDM09].

In this paper we exploit the connection between the optimal assignment problem and entropy maximization. The latter is well studied in the field of convex optimization [FRT97].

The main idea is to think of the optimal assignment problem as the limit of a deformation of an entropy maximization problem. More precisely, given an  $n \times n$  nonnegative matrix  $A = (a_{ij})$ , let us look for an  $n \times n$  bistochastic matrix  $X = (x_{ij})$  maximizing the relative entropy

$$J_p(X) := - \sum_{1 \leq i, j \leq n} x_{ij} (\log(x_{ij}/a_{ij}^p) - 1) , \quad (1.1)$$

---

\* INRIA Saclay – Île-de-France & Centre de Mathématiques appliquées, Ecole Polytechnique, 91128 Palaiseau, France ([Meisam.Sharify@inria.fr](mailto:Meisam.Sharify@inria.fr)).

<sup>†</sup>INRIA Saclay – Île-de-France & Centre de Mathématiques appliquées, Ecole Polytechnique, 91128 Palaiseau, France ([Stephane.Gaubert@inria.fr](mailto:Stephane.Gaubert@inria.fr)).

<sup>‡</sup>INRIA Saclay – Île-de-France & Laboratoire de Recherche en Informatique, Université Paris-Sud 11, France ([Laura.Grigori@inria.fr](mailto:Laura.Grigori@inria.fr)).

This work has been supported in part by the French National Research Agency (ANR) through the COSINUS program (projects PETAL no ANR-08-COSI-009 and PETALH no ANR-10-COSI-013).

Here,  $p$  is the deformation parameter. We will show in Section 2 that when  $p$  goes to infinity, the unique solution  $X(p) = (x_{ij}(p))$  of the entropy maximization problem converges to a point  $X(\infty)$  which is of maximal entropy among the ones in the convex hull of the matrices representing optimal permutations. In particular, if there is only one optimal permutation,  $X(p)$  converges to the matrix representing this optimal permutation. In Section 3 we prove that, for  $X(p)$  as the solution to equation (1.1) for some value of  $p$  and for  $X(\infty)$  as the solution when  $p \rightarrow \infty$ , we have

$$|x_{ij}(p) - x_{ij}(\infty)| = O(\exp(-cp))$$

for  $c > 0$ . This shows an exponential convergence to the optimal solution when  $p$  increases.

The maximal entropy matrix  $X(p)$  can be computed by any matrix scaling algorithm such as Sinkhorn iteration [SK67] or Newton method [KR07]. Subsequently, these iterative methods can be used to develop new algorithms to solve the optimal assignment problem and related combinatorial optimization problems.

An interesting application of this new approach, is the solution of large scale dense optimal assignment problems. Several efforts have been made to solve this problem [BT09, LO94]. A well-known application arises from the approximation algorithms and heuristics for solving the Asymmetric Traveling Salesman Problem or the Vehicle Routing Problem. There are also some applications in object recognition and computer vision. An application to cosmology (reconstruction of the early universe) can be found in the work of Brenier et al. [BFH<sup>+</sup>03]. For a survey on the applications of large dense linear assignment problems, we refer the reader to [BT09]. Models of large dense random assignment problems are also considered in [MPV, Ch. VII] from the point of view of statistical physics.

Here, we investigate a preprocessing algorithm which can be used to solve large scale optimal assignment problems. This preprocessing is based on an iterative method that eliminates the entries not belonging to an optimal assignment. This reduces the initial problem to a much smaller problem in terms of memory requirements. This is illustrated in Figures 1.1 and 1.2.

The idea of this algorithm is to take  $p$  large enough, then apply a diagonal scaling algorithm to  $A^{(p)}$  until convergence to a bistochastic matrix  $X$ , and finally delete the small entries of  $X$ . Here, the exponential of  $A^{(p)}$  leads to numerical overflow for large values of  $p$ . However, we shall show that it is possible to implement this iteration in a numerically stable way. The present algorithm assumes the existence of at least one matching, since otherwise, the Sinkhorn iteration may not converge. However, we note that matrix balancing (Sinkhorn iteration) can also be used to detect the existence of a perfect matching, as shown by Linial, Samorodnitsky and Wigderson [LSW00].

We consider two variants of the algorithm, one by using the Sinkhorn iteration as the diagonal scaling algorithm and the other one by using Newton iteration. The advantage of Sinkhorn iteration is that, it can be efficiently implemented in parallel [ADRU08, DRU08]. Thus we show that for very large dense optimal assignment problems the data of which cannot be stored in one machine, the parallel Sinkhorn iteration can be used to reduce the size of the problem and then, it can be solved by any classical method. On the other hand, the advantage of Newton method is the speed of the convergence to bistochastic matrix.

For both variants, we present several numerical results of various full and sparse matrices from gallery of Matlab and *The University of Florida Sparse Matrix Collection*. We show that the Sinkhorn iteration can be efficiently used to decrease the

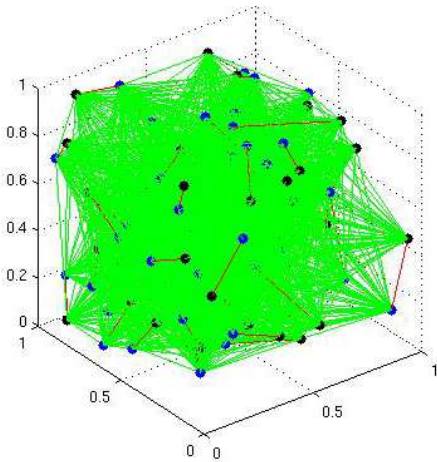


FIG. 1.1. *Euclidean random assignment problem (Section 4.1.2)*

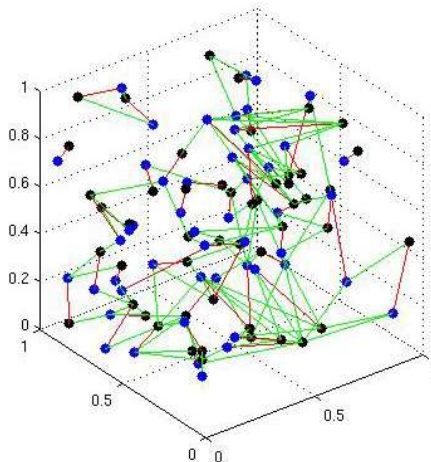


FIG. 1.2. *Reduced problem*

size of the dense matrices, up to 99% in a small number of iterations. For Newton iteration, we show that it is not only efficient for dense matrices but also efficient for sparse symmetric matrices.

Note also that the present approach yields approximate dual variables, which provide an approximate optimality certificate for the assignment which is found (Section 4.1.1).

In the last section, we introduce an iterative method which is based on a modification of the Sinkhorn scaling algorithm, in which the deformation parameter is slowly increased (this procedure is reminiscent from simulated annealing, the parameter  $p$  playing the role of the inverse of the temperature). We prove that this iteration, which we refer to as *deformed-Sinkhorn iteration*, converges to a matrix whose entries that belong to the optimal permutations are nonzero, while all the other entries are zero. An estimation of the rate of convergence is also presented, but this appears to be mostly of theoretical interest since in practice, the convergence of this variant appears to be slow.

**2. Entropy maximization and matrix scaling.** The diagonal scaling problem can be generally defined as finding diagonal matrices  $D_r$  and  $D_c$  with positive diagonal entries such that the scaled matrix  $D_r A D_c$  has prescribed row and column sums. Due to the variety of its applications, this problem has been well studied [MS69, Bru74, SK67]. A comparison of the proposed algorithms to solve this problem, can be found in [SZ90]. A remarkable special case arises when the row and column sums of the matrix  $X = D_r A D_c$  are required to be identically one, so that  $X$  is bistochastic. Then, the following theorem provides a sufficient condition for the existence of a diagonal scaling.

**THEOREM 2.1** (Sinkhorn [SK67]). *Let  $A$  be an  $n \times n$  nonnegative matrix with total support (every positive entry belongs to a diagonal). Then there exist diagonal matrices  $D_r$  and  $D_c$  such that  $D_r A D_c$  is bistochastic. Moreover, if  $A$  is fully indecomposable, then  $D_r$  and  $D_c$  are unique up to a constant factor.*

Now, consider the following optimization problem, which consists in finding an

$n \times n$  bistochastic matrix  $X = (x_{ij})$  maximizing the following relative entropy

$$\max_{X \in B_n} J_p(X), \quad J_p(X) := \sum_{ij} x_{ij} b_{ij} + p^{-1} S(X), \quad b_{ij} = \log a_{ij}, \quad (2.1)$$

where

$$S(X) := - \sum_{ij} x_{ij} \log x_{ij}$$

is the entropy function,  $p > 0$  is a parameter, and  $B_n$  denotes the set of  $n \times n$  bistochastic matrices. The convention  $0 \times (-\infty)$  is understood when interpreting the product  $x_{ij} b_{ij}$ .

We shall assume that the matrix  $A := (a_{ij})$  has total support, so that the diagonal matrices  $D_r$  and  $D_c$  are known to exist. We denote by  $G(A) := \{(i, j) \mid a_{ij} > 0\}$  the *pattern* (set of non-zero entries) of the matrix  $A$ .

The general relation between the entropy maximization and scaling problems is well known, see e.g. [Sch89] for an overview. We shall need in particular the following result.

**PROPOSITION 2.2** (Corollary of [BLN94, Th. 3.1]). *Let  $A$  be a matrix with total support. Then, the solution  $X(p)$  of the entropy maximization problem indicated in Equation 2.1 is unique and it is characterized by the existence of two positive vectors,  $U$  and  $V$ , such that  $x_{ij} = a_{ij}^p u_i v_j$  for all  $i, j$ .*

Thus, the characterization of the proposition shows that  $X$  is obtained from the  $p$ th Hadamard power  $A^{(p)} := (a_{ij}^p)$  by a diagonal scaling.

The previous proposition is a special case of Theorem 3.1 of [BLN94], which is established in a more general infinite dimensional setting (for  $p = 1$ ; but the result for an arbitrary  $p$  follows trivially from it). We shall need in the sequel a few elements of the proof, which we next include.

First, the function  $J_p$  is upper semi-continuous, and  $B_n$  is compact, hence, the maximum of  $J_p$  over  $B_n$  is attained. If there is at least one permutation  $\sigma$  such that  $\sum_i b_{i\sigma(i)} > -\infty$ , the associated permutation matrix  $X = (x_{ij})$ , with  $x_{ij} = 1$  if  $j = \sigma(i)$ , and  $x_{ij} = 0$  otherwise, is such that  $J_p(X) > -\infty$ . Then since the maximum of  $J_p$  is attained, its value must be finite. Moreover, since the objective function is strictly concave and the feasible set is convex, the point of maximum  $X(p)$  is unique.

We claim that  $X(p)$  has the same pattern (set of positions of non-zeros entries) as the matrix  $A$ .

To see this, let  $Y$  be a bistochastic matrix with the same pattern as  $A$ , i.e.  $y_{ij} > 0$  iff  $a_{ij} > 0$ . Assume by contradiction that  $X(p)$  does not have the same pattern as  $A$ , so that  $x_{ij}(p) = 0$  and  $y_{ij}(p) > 0$  for some  $(i, j)$ . Then because the right derivative of the function  $t \mapsto -t \log t$  at  $t = 0^+$  is infinite, the right derivative of  $t \mapsto J_p(X(p) + t(Y - X(p)))$  at  $t = 0^+$  is easily seen to be infinite, and so,  $J_p(X(p) + t(Y - X(p))) > 0$  and  $X(p) + t(Y - X(p)) \in B_n$  hold for  $t$  small enough, contradicting the optimality of  $X(p)$ . Hence, the claim is established.

Consider now the Lagrange function

$$L(X, U, V) = J_p(X) + \sum_i u_i \left( \sum_j x_{ij} - 1 \right) + \sum_j v_j \left( \sum_i x_{ij} - 1 \right),$$

where  $U = (u_i)$  and  $V = (v_j)$  are vectors of Lagrange multipliers. The stationarity condition implies that if  $X$  is an optimal solution of the entropy maximization problem

indicated in Equation 2.1, then there must exist two vectors of multipliers  $U$  and  $V$  such that, for all  $(i, j) \in G(A)$ ,

$$\frac{\partial L}{\partial x_{ij}} = b_{ij} - p^{-1}(1 + \log x_{ij}) + u_i + v_j = 0 .$$

It follows that

$$x_{ij}(p) = \exp(p(b_{ij} + u_i + v_j) - 1) , \quad \forall (i, j) \in G(A)$$

showing that  $X$  is obtained from the  $p$ th Hadamard power  $A^{(p)} := (a_{ij}^p)$  by a diagonal scaling.

Using the latter characterization of  $X(p)$ , we observe that:

$$J_p(X(p)) = - \sum_i \log u_i - \sum_j \log v_j .$$

We now study the convergence of  $X(p)$  as  $p$  tends to infinity. We shall consider the face  $F$  of the polytope of bistochastic matrices consisting of the optimal solutions of the linear programming formulation of the optimal assignment problem

$$\max_{x \in B_n} \sum_{ij} x_{ij} b_{ij} = \max_{\sigma \in \mathfrak{S}_n} \sum_i b_{i\sigma(i)} .$$

**THEOREM 2.3.** *As  $p$  tends to infinity, the matrix  $X(p)$  converges to the unique matrix  $X^*$  maximizing the entropy among the ones that belong to the face  $F$  consisting of the convex hull of optimal permutation matrices. In particular, if the solution of the optimal assignment problem is unique, then  $X(p)$  converges to the associated bistochastic matrix.*

*Proof.* Since  $X(p)$  is the point of maximum of  $J_p$ ,

$$\begin{aligned} J_p(X(p)) &= \sum_{ij} x_{ij}(p) b_{ij} + p^{-1} S(X(p)) \\ &\geq J_p(X^*) = \sum_{ij} x_{ij}^* b_{ij} + p^{-1} S(X^*) \\ &= \max_{\sigma \in \mathfrak{S}_n} \sum_i b_{i\sigma(i)} + p^{-1} S(X^*) \end{aligned}$$

Consider a sequence  $(p_k)_{k \geq 1}$  converging to infinity, and assume that  $X(p_k)$  converges to some matrix  $Z$ , which must belong to  $B_n$ . Setting  $p = p_k$  in the previous inequality and taking the limit as  $k$  tends to infinity, we get  $\sum_{ij} z_{ij} b_{ij} \geq \max_{\sigma \in \mathfrak{S}_n} \sum_i b_{i\sigma(i)}$ , which shows that  $Z$  belongs to the face  $F$ .

Observe that

$$p_k^{-1}(S(X(p_k)) - S(X^*)) = (J_{p_k}(X(p_k)) - J_{p_k}(X^*)) + \left( \sum_{ij} x_{ij}^* b_{ij} - \sum_{ij} x_{ij}(p_k) b_{ij} \right)$$

is the sum of two nonnegative terms, because  $X(p_k)$  is a point of maximum of  $J_{p_k}$ , and  $X^* \in F$  is a convex hull of matrices representing optimal permutations. It follows that  $S(X(p_k)) - S(X^*) \geq 0$ , and so, if  $Z$  is any accumulation point of  $X(p_k)$  as  $k$

tends to infinity,  $S(Z) - S(X^*) \geq 0$ , showing that  $Z$  is of maximal entropy among the matrices in  $F$ . Since the entropy function is strictly convex,  $X^*$  is the only point with the latter property, and so every accumulation point of  $X(p_k)$  is equal to  $X^*$ , showing that  $X(p)$  converges to  $X^*$  as  $p \rightarrow \infty$ .  $\square$

**COROLLARY 2.4.** *If there is only one optimal permutation, then  $X(p)$  converges to the corresponding permutation matrix.*

**3. The speed of convergence.** We have already shown in Theorem 2.3 that the maximal entropy solution  $X(p)$  converges as  $p$  tends to infinity, to a matrix  $X(\infty)$  which is a convex hull of optimal permutation matrices. In particular,  $X(p)$  converges to an optimal permutation matrix if the optimal permutation is unique. Now, the question is how fast is this convergence. This is answered by the following theorem.

**THEOREM 3.1.** *Assume that the matrix  $A$  has total support and that  $\log a_{ij} \in \mathbb{Q}$ , for all  $(i, j)$  such that  $a_{ij} > 0$ . Then, there exists a positive constant  $c$  such that, for all  $i, j \in [n]$ ,*

$$|x_{ij}(p) - x_{ij}(\infty)| = O(\exp(-cp))$$

To establish Theorem 3.1, recall that a real Puiseux series in the variable  $t$  is an expression of the form

$$f = \sum_{k \geq \bar{k}} c_k t^{k/r} \quad (3.1)$$

where  $r \in \mathbb{N}$  is positive,  $\bar{k} \in \mathbb{Z}$ ,  $c_k \in \mathbb{R}$  for all  $k$ , and the sum is taken over all  $k \in \mathbb{Z}$  such that  $k \geq \bar{k}$ . We denote by  $\mathbb{R}_{\text{cvg}}\{\{t\}\}$  the set of real Puiseux series that are absolutely convergent for all  $t$  of small enough positive modulus.

**LEMMA 3.2.** *For all  $i, j \in [n]$ , there exists a Puiseux series of the form (3.1), such that*

$$x_{ij}(p) = f(\exp(-p)) = \sum_{k \geq \bar{k}} c_k \exp(-pk/r)$$

*the latter series being absolutely convergent for all large enough  $p$ .*

In order to establish this result, we shall use some tools from the theory of real ordered fields, for which we refer the reader to [BPR06, chapter 2].

Let us consider the following statement: if a nonnegative matrix  $A$  has total support, then there exists a unique nonnegative matrix  $X$  with row and column sums 1, and there exist diagonal matrices  $D$  and  $D'$  with positive diagonal entries such that

$$A = DXD'.$$

According to Sinkhorn's theorem [SK67] and to Proposition 2.2, this statement is true when the entries of  $A, X, D, D'$  belong to the field of real numbers. Moreover, this statement belongs to the first order theory of the real closed field  $(\mathbb{R}, +, \times, 0, 1, >)$ . By Tarski's theorem [Tar51], any first order statement that is valid in a special real closed field must also be valid in any real closed field. In particular, the above statement holds over the field of convergent real Puiseux series, which is known to be a real closed field. Indeed, the fact that formal Puiseux series constitute a real closed field is standard, the proof that the same is true in the case of convergent Puiseux series can be found in [BK76, § 10].

Thus for a matrix  $A(t) \in \mathbb{R}_{\text{cvg}}\{\{t\}\}^{n \times n}$  with total support, there exists diagonal matrices  $D(t), D'(t) \in \mathbb{R}_{\text{cvg}}\{\{t\}\}^{n \times n}$  together with a unique bistochastic matrix  $\hat{X}(t) \in \mathbb{R}_{\text{cvg}}\{\{t\}\}^{n \times n}$  such that  $A(t) = D(t)\hat{X}(t)D'(t)$ .

We choose now the matrix  $A(t) = (a_{ij}(t))$  such that  $a_{ij}(t) = t^{\log a_{ij}}$  where  $\log a_{ij} \in \mathbb{Q}$ . Then, the entries of the corresponding matrix  $\hat{X}(t)$  have the form

$$\hat{x}_{ij}(p) = \sum_{k=\bar{k}_{ij}}^{+\infty} c_{ijk} t^{k/r_{ij}}$$

and this series is convergent for a suitably small positive  $t$ . Make now the substitution  $t = \exp(-p)$ . We deduce that for all suitably large  $p$ ,

$$x_{ij}(p) = \sum_{k=\bar{k}_{ij}}^{+\infty} c_{ijk} \exp(-pk/r_{ij}) . \quad (3.2)$$

Since  $x(p)_{ij}$  has a finite limit as  $p \rightarrow \infty$ , understanding that  $\bar{k}_{ij}$  is the first index  $k$  for which the coefficient  $c_{ijk}$  is non-zero, we necessarily have  $\bar{k}_{ij} \geq 0$ , so that  $x_{ij}(\infty)$  can be identified to the constant term in the latter series. Setting  $c = \min_{i,j} (\bar{k}_{ij} + 1)/r_{ij}$  we get

$$|x_{ij}(p) - x_{ij}(\infty)| = O(\exp(-cp)) ,$$

which proves Theorem 3.1.

REMARK 3.3. The assumption that  $\log a_{ij} \in \mathbb{Q}$  in Theorem 3.1 is inconvenient. It could be avoided by replacing the field of converging Puiseux series by a field of converging generalized Dirichlet series, along the lines of [Mar]. However, this would require working out the convergence issues which are not treated in [Mar].

REMARK 3.4. The formulation (2.1) is somehow reminiscent of interior point methods, in which the entropy  $S(X) = -\sum_{ij} x_{ij} \log x_{ij}$  is replaced by a log-barrier function (the latter would be  $\sum_{ij} \log x_{ij}$  in the present setting). The present  $X(p)$  thought of as a function of  $p \rightarrow \infty$  is analogous to the *central path*, and as does the central path,  $X(p)$  converges to a face containing optimal solutions. However, the entropy  $S(X)$  does not satisfies the axioms of the theory of self-concordant barriers on which the analysis of interior point methods is based. Indeed, the speed of convergence in  $O(\exp(-cp))$  appears to be of a totally different nature by comparison with the speed of  $O(1/p)$  observed in interior point methods [NN94].

EXAMPLE 3.5. The constant  $c$  appearing in Theorem 3.1 can be small if there are several nearly optimal permutations, and then a large value of  $p$  may be needed to approximate  $X(\infty)$ . However, in such cases, a much smaller value of  $p$  turns out to be enough for the method described in the next sections, the aim of which is to eliminate a priori entries not belonging to (nearly) optimal permutations. This is illustrated by the following matrix, in which the identity permutation is optimal, and the transposition  $(1, 2)$  is nearly optimal:

$$A = \begin{pmatrix} 1 & 0.99 & 0.99 \\ 0.99 & 1 & 1/3 \\ 0.25 & 0.5 & 1 \end{pmatrix} .$$

For  $p = 10$ , we have the following matrix, the significant entries of which indicate

precisely the optimal and nearly optimal permutations:

$$\begin{pmatrix} 0.5195148 & 0.4595136 & 0.0210196 \\ 0.4804643 & 0.5195864 & 0.0000004 \\ 0.0000209 & 0.0209000 & 0.9789800 \end{pmatrix}.$$

The convergence of  $X(p)$  to  $X(\infty)$  is illustrated in Figures 3.1. Observe that the graph of  $\log x_{ij}(p)$  as a function of  $p$  is approximately piecewise affine. In fact, each piece corresponds to a monomial in the Puiseux series expansion (3.2) (see [Vir01] for an explanation of this fact). The path  $p \mapsto X(p)$  converges quickly to the face containing the two nearly optimal permutations and slowly to the unique optimal permutation.

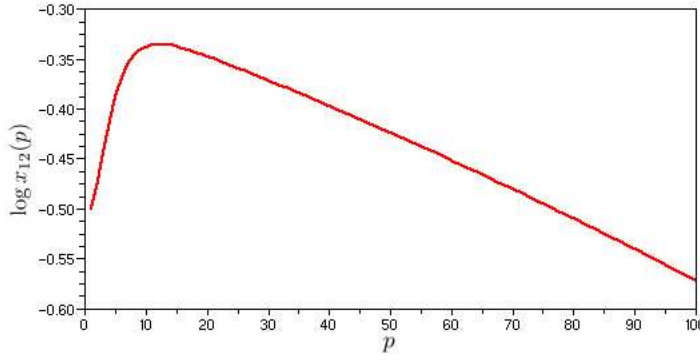


FIG. 3.1. The variation of  $\log_{10} x_{12}(p)$  as a function of  $p$ .

REMARK 3.6. Finding an explicit formula for the speed of convergence  $c$  appears to be an interesting combinatorial problem (which is beyond the scope of this paper).

**4. Preprocessing for the optimal assignment problem.** For a fixed  $p > 0$ , the solution for the entropy maximization problem displayed in Equation (2.1) can be computed by any scaling algorithm such as Sinkhorn iteration or Newton method. Using Theorem 3.1, it can be seen that if the original matrix has only one optimal permutation, the order of magnitude of all the entries which belong to the optimal permutation will be  $1 \pm O(\exp(-cp))$  while the order of magnitude of all other entries are  $O(\exp(-cp))$ . As an example, consider the following 5 by 5 random matrix with the bold entries belonging to optimal permutation.

$$A = \begin{pmatrix} 0.292 & 0.502 & \mathbf{0.918} & 0.281 & 0.686 \\ 0.566 & \mathbf{0.437} & 0.044 & 0.128 & 0.153 \\ 0.483 & 0.269 & 0.482 & \mathbf{0.778} & 0.697 \\ 0.332 & 0.633 & 0.264 & 0.212 & \mathbf{0.842} \\ \mathbf{0.594} & 0.405 & 0.415 & 0.112 & 0.406 \end{pmatrix}$$

By applying the Sinkhorn iteration on  $A^{(50)}$  the following matrix can be computed.

$$X(50) = \begin{pmatrix} 3.4E-27 & 1.5E-08 & \mathbf{1.0E+00} & 7.4E-26 & 4.7E-06 \\ 4.8E-02 & \mathbf{9.4E-01} & 4.6E-56 & 4.0E-32 & 7.9E-28 \\ 2.5E-13 & 4.6E-19 & 9.3E-12 & \mathbf{1.0E+00} & 1.0E-02 \\ 1.5E-23 & 1.2E-02 & 6.2E-27 & 4.3E-31 & \mathbf{9.8E-01} \\ \mathbf{9.5E-01} & 4.1E-02 & 6.2E-07 & 1.0E-34 & 2.3E-06 \end{pmatrix}$$

Thus, for sufficiently large values of  $p$ , when  $X(p)$  is an  $\epsilon$ -bistochastic matrix, meaning that some distance between  $X(p)$  and a bistochastic matrix is less than  $\epsilon$ , one may delete all the small entries which are less than a threshold  $t$ , chosen consistent with  $\epsilon$ , while keeping all others. In this way the size of the original problem in terms of memory requirements will be reduced to a much smaller one.

For a column (row) stochastic matrix, that is a matrix for which the sum of all columns (rows) are one, the distance to the set of bistochastic matrices will be measured by  $\max_i |r_i - 1|$  where  $r_i$  indicates the  $i$ th row (column) sum.

Determining the coarsest accuracy  $\epsilon$  and the maximal threshold  $t$  which are needed to find an optimal permutation would require to know the maximal entropy solution  $X(\infty)$  characterized in Theorem 2.3. This information is in general not available. However, the worst case can be considered to be the one where  $X(\infty)$  is uniform, with all entries equal  $1/n$  (and  $n!$  optimal permutations). Since we need to preserve the optimal permutations, this leads to a conservative choice  $\epsilon = t = 1/n$  which we adopted in the present experimental results. The choice of the value of  $p$  will be discussed in Section 4.1.2. This leads to Algorithm 1.

---

**Algorithm 1** An optimal assignment preprocessing for fixed  $p$ 


---

```

input:  $A, p$ 
 $n \leftarrow \text{size}(A, 1)$ 
 $\epsilon, t \leftarrow 1/n$ 
comment: Prescaling
if  $\frac{\max(A)}{\min(A)} > e$  then
     $m \leftarrow \frac{1}{\log(\max(A)/\min(A))}$ ,  $c \leftarrow e^{\frac{\log(\min(A))}{\log(\max(A)/\min(A))}}$ 
     $A \leftarrow \frac{1}{c} A^{(m)}$ 
else
     $A \leftarrow \frac{1}{\min(A)} A$ 
end if
 $B \leftarrow A^{(p)}$ 
comment: Main loop
repeat
    apply one iteration of any diagonal scaling algorithm to  $B$  so  $B \leftarrow DB'D$ , where
     $D, D'$  are diagonal matrices
until  $B$  is  $\epsilon$ -bistochastic
Delete all the entries of  $B$  which are less than a threshold  $t$ 

```

---

The naive computation of  $A^{(p)}$  is numerically unstable for large values of  $p$ . This can be avoided by the prescaling step in Algorithm 1. Then, we set  $\max(A) = \max_{ij} a_{ij}$ ,  $\min(A) = \min_{a_{ij} > 0} a_{ij}$ . By applying this prescaling, all the nonzero scaled entries will be placed in the  $[1, e]$  interval. In the case when  $\max(A)/\min(A) > e$ , the prescaling has another interesting property, that is, the scaled matrix is invariant by any entrywise power of the input matrix. In other words, if we apply the prescaling to the matrix  $A^{(q)}$ , for all  $q \geq 1$ , the matrix obtained after the prescaling step turns out to be independent of the choice of  $q$ . When  $\frac{\max(A)}{\min(A)} < e$  the entries of  $A$  have already been located in the interval  $\min(A)[1, e]$ , then we do not need to perform the previous prescaling since the denominator in the formula defining  $m$  will be small if  $\max(A)$  is close to  $\min(A)$ . We shall also see in Section 4.1.1 that the iterations can be implemented robustly for large values of  $p$  by working with log-coordinates. Next,

we provide more details on the proposed algorithm.

**4.1. Sinkhorn iteration.** A simple way to compute the diagonal matrices  $D, D'$  is Sinkhorn iteration [SK67]. This algorithm starts from a given matrix  $A$ , divides every row by its sum, then every column of the new matrix by its sum, and so on, until the matrix obtained in this way converges to a bistochastic matrix. The advantage of this algorithm is that, it can be efficiently implemented in parallel [ADRU08] and it can be applied to any non-negative matrix which has at least one nonzero permutation. The disadvantage is that, it is generally slower than other methods.

Recall first that the open cone  $C = \{x \in \mathbb{R}^n : x_i > 0, \forall i\}$  consisting of positive vectors of  $\mathbb{R}^n$  is equipped with Hilbert's projective metric, defined by

$$d(x, x') = \log \max_{i,j} \frac{x_i x'_j}{x'_i x_j}$$

Note that  $d(x, x')$  is zero if and only if the vectors  $x$  and  $x'$  are proportional. We refer to [BR97, § 6] for more background. In particular, if  $A$  is a positive matrix, a theorem of Birkhoff shows that the map  $x \mapsto Ax$  is a contraction in Hilbert's projective metric, with a contraction rate

$$\kappa(A) := \sup \left\{ \frac{d(Ay, Ay')}{d(y, y')} : y, y' \in C, y, y' \text{ non proportional} \right\} = \frac{\theta(A)^{1/2} - 1}{\theta(A)^{1/2} + 1} ,$$

where

$$\theta(A) = \exp \sup \{d(Ay, Ay') : y, y' \in C\} = \max_{i,j,p,l} \frac{a_{ir} a_{jl}}{a_{jr} a_{il}}$$

The following result is a consequence of this theorem.

**PROPOSITION 4.1** (Franklin and Lorenz [FL89]). *For a positive matrix  $A$ , the global rate of convergence of Sinkhorn iteration is bounded above by  $\kappa(A)^2$ .*

This general bound is applicable only for positive matrices and it can be coarse in practice. Recently, Knight [Kni08] provided a local rate of convergence. Due to his work, for classical Sinkhorn iteration the local rate of convergence of a fully indecomposable matrix, is bounded by  $\sigma_2^2$  where  $\sigma_2$  is the second singular value of the bistochastic matrix to which the iteration converges. Hence, the following result allows us to estimate the local convergence rate of Sinkhorn iteration, as  $p \rightarrow \infty$ .

**PROPOSITION 4.2.** *Assume that there is only one optimal permutation. Then, there is a constant  $c > 0$  such that*

$$1 - O(\exp(-cp)) \leq \sigma_2(X(p)) \leq 1 \quad \text{as } p \rightarrow \infty$$

*Assume now that the matrix  $X(\infty)$  is fully indecomposable (which implies that there are several optimal permutations). Then,*

$$\sigma_2(X(p)) \rightarrow \sigma_2(X(\infty)) < 1 \quad \text{as } p \rightarrow \infty .$$

*Proof.* Due to the perturbation theorem of Mirsky [Mir60], for any unitarily invariant norm  $\|\cdot\|$  and  $n \times n$  matrices,  $X$  and  $\tilde{X}$  with singular values  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p$  and  $\tilde{\sigma}_1 \geq \tilde{\sigma}_2 \geq \dots \geq \tilde{\sigma}_p$  respectively, we have,

$$\|\text{diag}(\tilde{\sigma}_i - \sigma_i)\| \leq \|\tilde{X} - X\|.$$

So, for  $X(p)$  and  $X(\infty)$ ,

$$|\sigma_2(X(p)) - \sigma_2(X(\infty))| \leq \|X(p) - X(\infty)\|_2 \leq O(\exp(-cp))$$

for which the constant  $c$  depends on the coefficients of the Puiseux series and possibly on the dimension of  $X(p)$ . Thus, if the original matrix has only one optimal permutation,  $\sigma_2(X(\infty)) = 1$  which implies that

$$1 - O(\exp(-cp)) \leq \sigma_2(X(p))$$

Moreover according to the Birkhoff-von Neumann theorem [Bir46], for any norm  $\|\cdot\|$  on  $\mathbb{R}^n$  which is invariant under permutation of the coordinates and for any bistochastic matrix  $X$ ,  $\|X\| = 1$  and subsequently

$$1 - O(\exp(-cp)) \leq \sigma_2(X(p)) \leq 1$$

When  $X(\infty)$  is fully indecomposable, since the multiplication of two fully indecomposable matrices is also fully indecomposable,  $M = X(\infty)X^T(\infty)$  is fully indecomposable. Note also that for all  $1 \leq i \leq n$ ,  $m_{ii} = \sum_{j=1}^n x_{ij}^2 > 0$ , which implies that  $M$  is primitive. Then, according to the Perron-Frobenius theorem, all the eigenvalues of  $M$  distinct from  $\rho(M)$  have a modulus strictly smaller than  $\rho(M) = 1$  which yields  $\sigma_2(X(\infty)) < 1$ .  $\square$

**4.1.1. Logarithmic  $p$ -Sinkhorn iteration and approximate optimality certificates.** As it was discussed before, computing the  $p$ th Hadamard power of  $A$  may cause some numerical difficulties. To avoid this problem a prescaling has been proposed, after which all the matrix entries are in  $[1, e]$  interval. A theoretical disadvantage of this prescaling is that the increase of  $p$  is limited since  $e^p < l$ , where  $l$  is the largest number, in the numerical range. However, we next give a log-coordinate implementation of Sinkhorn iteration which avoids this limitation. This will provide as a by product a certificate allowing one to check the approximate optimality of a permutation.

Let  $A \in \mathbb{R}^{n \times n}$  be a real non-negative matrix which has total support. For a given  $p$ , consider the following iteration for a sequence of vectors  $U_k, V_k \in \mathbb{R}^n$

$$V_0 = \mathbb{1} \tag{4.1}$$

$$U_{k+1} = \mathcal{I}(A^{(p)} V_k) \tag{4.2}$$

$$V_{k+1} = \mathcal{I}(A^{(p)T} U_{k+1}) \tag{4.3}$$

where  $\mathbb{1}$  is a vector  $[1, 1, \dots, 1]^T$  of dimension  $n$  and  $\mathcal{I}$  is an operator which inverts the entries of a vector.

**PROPOSITION 4.3.** *For a nonnegative matrix,  $A$ , which has total support, the iteration defined by Equations 4.1, 4.2 and 4.3 coincides with Sinkhorn iteration.*

*Proof.* Let  $W_k$  and  $Z_k$  respectively, be column scaled and row scaled matrices defined as the following:

$$\begin{aligned} W_k &= \text{diag}(U_k) A^{(p)} \text{diag}(V_k) \\ Z_k &= \text{diag}(U_{k+1}) A^{(p)} \text{diag}(V_k) \end{aligned}$$

Also, let  $\mathcal{C}$  denote the column scaling operator in which all the columns of a matrix are divided by it's sums and  $\mathcal{R}$  be the similar operator for rows. It is easy to verify

that,  $\mathcal{R}(DB) = \mathcal{R}(B)$  and  $\mathcal{C}(BD) = \mathcal{C}(B)$  for any diagonal matrix  $D$ . According to the definition

$$Z_k = \mathcal{R}(A^{(p)} \text{diag}(V_k)) = \mathcal{R}(\text{diag}(U_k) A^{(p)} \text{diag}(V_k)) = \mathcal{R}(W_k)$$

a similar statement can be proved for  $W_k$ , that is,  $W_K = \mathcal{C}(Z_K)$  which completes the proof.  $\square$

Assume that  $\bar{U}_k = (u_i^k) = p^{-1} \log U_k$  and  $\bar{V}_k = (v_i^k) = p^{-1} \log V_k$ , then, the logarithmic form of this iteration can be written as:

$$\begin{aligned}\bar{u}_i^{k+1} &= -\frac{1}{p} \log \sum_j \exp p(\log a_{ij} + \bar{v}_j^k) \\ \bar{v}_i^{k+1} &= -\frac{1}{p} \log \sum_j \exp p(\log a_{ji} + \bar{u}_j^{k+1})\end{aligned}$$

Let

$$\begin{aligned}\hat{x}_{ij} &= \log a_{ij} + \bar{v}_j^k - \max_j (\log a_{ij} + \bar{v}_j^k) \\ \hat{y}_{ji} &= \log a_{ji} + \bar{u}_j^{k+1} - \max_j (\log a_{ji} + \bar{u}_j^{k+1})\end{aligned}$$

for which  $\hat{x}_{ij}, \hat{y}_{ji} \leq 0$ . The logarithmic iteration can be reformulated by using  $\hat{x}_{ij}$  and  $\hat{y}_{ji}$  as the following:

$$\bar{u}_i^{k+1} = -\max_j (\log a_{ij} + \bar{v}_j^k) - \frac{1}{p} \log \sum_j \exp p \hat{x}_{ij} \quad (4.4)$$

$$\bar{v}_i^{k+1} = -\max_j (\log a_{ji} + \bar{u}_j^{k+1}) - \frac{1}{p} \log \sum_j \exp p \hat{y}_{ji} \quad (4.5)$$

The last iteration can be computed for a sufficiently large  $p$ , without having numerical difficulties. We note that a related trick was used by Malajovich and Zubelli [MZ01] in a different context.

**PROPOSITION 4.4** (Approximate optimality certificate). *Let  $\bar{U}, \bar{V}$  and  $\hat{X}$  be produced by the  $p$ -Sinkhorn iteration. Also, let  $\zeta_i := \frac{1}{p} \log \sum_j \exp p \hat{x}_{ij}$  and let  $\text{Val}(\text{OAP})$  denote the logarithmic of the value of an optimal permutation. Then,*

$$\text{Val}(\text{OAP}) \leq -\sum_{i=1}^n \bar{u}_i - \sum_{j=1}^n \bar{v}_j - \sum_{i=1}^n \zeta_i . \quad (4.6)$$

*Proof.* Observe that at each step of the Sinkhorn iteration:

$$\log a_{ij} + \bar{v}_j^k \leq -\bar{u}_i^{k+1} - \zeta_i , \quad 1 \leq i \leq n$$

Let  $\sigma$  denote an optimal permutation. Choosing  $j = \sigma(i)$  in the previous inequality, and summing over  $1 \leq i \leq n$ , we get (4.6).  $\square$

In practice, this proposition will be used to check the validity of the preprocessing, by comparing the logarithm of the value of the permutation which is eventually found with the upper bound (4.6).

**4.1.2. Experimental results.** The experiments which are presented here have been obtained by using Sinkhorn iteration in Algorithm 1 as a diagonal scaling method. We used Matlab version 7.10.0. The detailed Matlab implementation of the algorithm is presented below.

Finding the best value for  $p$  seems to be tricky since increasing  $p$  yields a slow convergence and at the same time, it yields the lower percentage of remaining entries. This fact also can be seen in Figures 4.1, 4.2 which illustrate the percentage of the remaining entries and the required number of Sinkhorn iterations, for several values of  $p$  for the “lotkin” 1000 by 1000 matrix from the gallery of Matlab.

---

Matlab code for p-Sinkhorn iteration

---

```
function [it,A]=psinkhorn(A)
    n=size(A,1);
    t=1/n;
    p=100;
    Min=min(A(A>0));
    Max=max(A(A>0));
    if (Max/Min)>exp(1) %prescaling
        m=1/(log(Max)-log(Min));
        c=exp(log(Min)/(log(Max)-log(Min)));
        A=(1/c)*(A.^m);
    else
        m=1/log(Max);
        A=A.^m;
    end
    A=A.^(p);
    d=(1/n)+1;
    it=0;
    while (d> 1/n) %main loop
        A=diag(sparse((A*ones(n,1)).^(-1)))*A;
        A=A*diag(sparse((A'*ones(n,1)).^(-1)));
        d=max(abs(sum(A')-1));
        it=it+1;
    end;
    [indx,indy]=find(A>t);
    A=sparse(indx,indy,1,n,n).*A;
end
```

---

In the following experiments, we set the parameter  $p$  to 100 which leads to a reasonable decrease in the size of the problem and generally does not yield to a slow convergence, however it could be any reasonably large value. Recall that the convergence is measured by  $\max_i |r_i - 1|$ , where  $r_i$  denotes the  $i$ th row (column) sum for a column (row) stochastic matrix. Table 4.1 displays the results for dense matrices from the gallery of test matrices of Matlab. For these experiments the dimension is 5000. The columns from left to right are: gallery name, number of nonzeros, number of iterations, the logarithmic value of optimal assignment and the percentage of remaining entries after deleting small entries. The same results are also presented for a random matrix, referred to as “rand”(the random function of

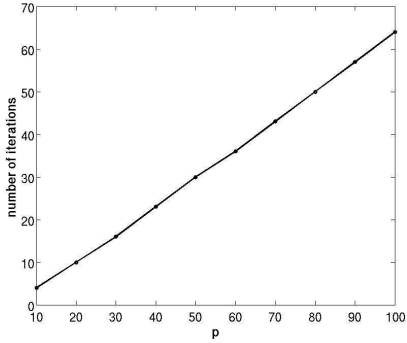


FIG. 4.1. The number of iterations as a function of  $p$ .

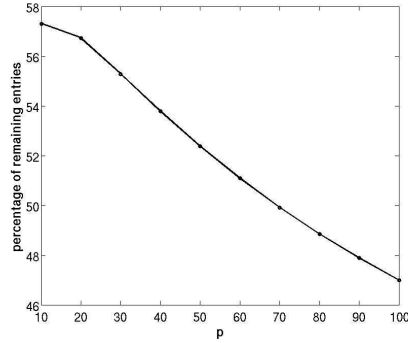


FIG. 4.2. The percentage of remaining entries as a function of  $p$ .

Matlab) and an Euclidean random matrix referred to as "Euclidean". The latter, which is of interest in statistical physics, is a matrix whose entries are functions of random points in an Euclidean space [Par02]. More precisely, we draw at random  $2n$  points  $x_1, \dots, x_n; y_1, \dots, y_n$  uniformly in the unit cube of  $\mathbb{R}^3$ . Then, we consider the matrix  $A = (a_{ij})$  where  $a_{ij} = \exp(-d(x_i, y_j))$  and  $d$  is the Euclidean distance. In this way, a permutation  $\sigma$  which maximizes  $\prod_{i=1}^n a_{ij}$  is the same permutation which minimizes the distance between these two sets of points.

TABLE 4.1

*Sinkhorn iteration for dense matrices from the gallery of test matrices of Matlab and for random and random Euclidean distance matrices*

Gallery	nnz	No. it.	Val(OAP)	Rem. En.(%)
cauchy	25000000	79	$4.54725E+00$	47.95
minij	25000000	473	$1.25025E+07$	26.57
moler	25000000	304	$4.99950E+07$	28.43
orthog	25000000	304	$4.99950E+07$	28.43
pei	25000000	1	$5.50000E+04$	00.02
prolate	25000000	42	$2.00000E+03$	00.66
randcorr	25000000	1	$5.00000E+03$	00.02
toeppd	25000000	1	$1.24767E+07$	00.02
chebvand	24997500	2	$5.00000E+03$	38.67
circul	25000000	1	$2.50000E+07$	19.48
cycl	25000000	3	$1.73422E+04$	13.23
lotkin	25000000	73	$5.54715E+00$	48.59
rand	25000000	2	$4.99837E+03$	28.38
Euclidean	25000000	417	$4.77693E+03$	01.49
chebspec	25000000	1084	$5.33411E+07$	01.98
lehmer	25000000	3537	$5.00000E+03$	18.58
gcdmat	25000000	11174	$1.25025E+07$	00.06

As Table 4.1 shows, For more than 58% of the cases, the algorithm converges very fast (in less than 80 iterations) and for 82% of the cases the algorithm converges in less than 500 iterations (which is less than 0.1 of the dimension of the input matrix). Also for more than 41% of the cases the original problem reduced to a new problem which has less than 2% of the original entries and in 82% it reduces to a new problem with less than 30% of the input entries. Since, the Sinkhorn iteration can be implemented in parallel, this method can be efficiently applied to large dense optimal assignment problems as a parallel preprocessing to reduce the size of the original problem.

We also tested several sparse matrices from *The University of Florida Sparse Matrix Collection*. The results show that using Sinkhorn iteration as a diagonal scaling

method in Algorithm 1 generally makes a slow convergence for sparse matrices.

**4.2. Newton iteration.** Solving the diagonal matrix scaling problem by using Newton iteration has been considered first in the work of Khachian and Kahalan-tari [KK92] for positive semidefinite symmetric matrices. They have considered the more general problem of finding a positive zero of the mapping

$$f(x) = b + Ax - x^{-1}$$

where  $A$  is a given matrix of dimension  $n$  and  $b$  is a fixed  $n$ -dimensional vector. They proposed a path-following Newton algorithm of complexity  $O(\sqrt{n}L)$  where  $L$  is the binary length of the input.

Recently, Knight and Ruiz have considered a Newton algorithm for nonnegative matrices [KR07]. For a symmetric matrix  $A$ , they considered the diagonal matrix scaling problem as finding a vector  $x$  such that

$$f(x) = D(x)Ax - \mathbb{1} = 0$$

where  $D(x) = \text{diag}(x)$ . If  $A$  is nonsymmetric, then the following matrix will be considered as the input of the algorithm.

$$S = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}$$

They showed that the Newton iteration can be written as

$$A_k x_{k+1} = Ax_k + D(x_k)^{-1} \mathbb{1}$$

where  $A_k = A + D(x_k)^{-1}D(Ax_k)$ . Thus in each iteration a linear system of equations should be solved for which they used the Conjugate Gradient method. In the nonsymmetric case, the latter linear system is singular, however it is proved that the system is consistent whenever  $A$  has support ( $A \geq 0$  has support if it has a positive diagonal). Our experiments which will be presented later shows that, the method works fast for dense nonsymmetric matrices. However with the default tuning parameters, it does not work fast in sparse nonsymmetric cases. More details and the exact implementation of this method can be found in [KR07]. Here, we used the later method in Algorithm 1 to find the scaling matrices. We also set the parameter  $p$  to 100 which is the same as Sinkhorn iteration.

In the following tables, No. it. denotes the total number of operations, each of them takes  $O(n^2)$  time to be done. This includes all the iterations of Conjugate Gradient method for each Newton step. Tables 4.2 and 4.3 show the results for dense symmetric and nonsymmetric matrices with dimension 5000. For both cases the algorithm converges rapidly in a small number of iterations. The percentage of the remaining entries is reasonably less than the original problem. In fact, in more than 38% of the cases, the original problem reduced to a much smaller problem which has less than 2% of the original entries and in 72% of the cases the problem reduces to a problem with less than 30% of the original entries.

Tables 4.4 and 4.5 show the result of this algorithm on several sparse symmetric and nonsymmetric matrices from *The University of Florida Sparse Matrix Collection*. These results show that the algorithm generally works very well for sparse symmetric matrices while the convergence for sparse nonsymmetric matrices is not fast.

TABLE 4.2  
Newton iteration for dense symmetric matrices

Gallery	nnz	No. it.	Val(OAP)	Rem. En.(%)
cauchy	25000000	156	$-4.10569E + 04$	47.95
fiedler	24995000	175	$3.91202E + 04$	35.73
gcdmat	25000000	152	$3.75911E + 04$	00.06
lehmer	25000000	166	$0.00000E + 00$	18.58
minij	25000000	167	$3.75911E + 04$	26.57
moler	25000000	167	$4.45149E + 04$	28.43
orthog	25000000	164	$-1.9561E + 04$	48.10
pei	25000000	151	$1.19895E + 04$	00.02
prolate	25000000	155	$-4.58145E + 03$	00.66
randcorr	25000000	151	$0.00000E + 00$	00.02
toeppd	25000000	151	$3.91132E + 04$	00.02

TABLE 4.3  
Newton iteration for dense nonsymmetric matrices

Gallery	nnz	No. it.	Val(OAP)	Rem. En.(%)
chebspec	25000000	251	$4.03274E + 04$	01.98
chebvand	24997500	166	$-1.19254E - 03$	38.67
circul	25000000	161	$4.25860E + 04$	19.48
cycl	25000000	162	$6.19386E + 03$	11.81
lotkin	25000000	257	$-4.10477E + 04$	48.59
rand	25000000	164	$-1.63137E + 00$	28.39
Euclidean	25000000	314	$-2.30779E + 02$	01.49

**5. Deformed Sinkhorn iteration.** In the previous section, we computed  $X(p)$  for a fixed value of  $p$ . However, it is natural to develop a “path following method” in which the value of  $p$  is gradually increased in the course of Sinkhorn balancing iterations. In this section we propose such an algorithm. We prove that if the matrix  $A$  has support ( $A$  has support if it has a positive diagonal), and if the growth of  $p$  is moderate enough, then the sequence of matrices produced by the algorithm converges to a point which belongs to the face generated by optimal permutations.

**5.1. Definition.** Let  $A \in \mathbb{R}^{n \times n}$  be a real non-negative matrix. Consider the following iteration which is a standard Sinkhorn iteration with a deformation of using a sequence  $p_m$  which goes to infinity.

$$\begin{aligned} U_{m+1} &= \mathcal{I}(A^{(p_{m+1})} V_m) \\ V_{m+1} &= \mathcal{I}(A^{(p_{m+1})T} U_{m+1}) \end{aligned}$$

Let  $W_{m+1}$  and  $Z_m$  respectively, be column scaled and row scaled matrices defined as the following:

$$\begin{aligned} W_{m+1} &= \text{diag}(U_{m+1}) A^{(p_{m+1})} \text{diag}(V_{m+1}) \\ Z_m &= \text{diag}(U_{m+1}) A^{(p_{m+1})} \text{diag}(V_m) \end{aligned} \tag{5.1}$$

**PROPOSITION 5.1.** *For a diagonal matrix  $D$ , real matrices  $B, C$  and the matrices  $W_m, Z_m$  in the iteration, the following properties hold.*

1.  $\mathcal{R}(C \circ (DB)) = \mathcal{R}(C \circ B)$  where  $\circ$  indicates the Hadamard product
2.  $W_m = \mathcal{C}(Z_{m-1})$
3.  $Z_m = \mathcal{R}(W_m \circ A^{(p_{m+1}-p_m)})$

TABLE 4.4  
Newton iteration for sparse symmetric matrices

Gallery	n	nnz	No. it.	Val(OAP)	Rem. En.(%)
2cubes_sphere	101492	1647264	155	$1.29645E+06$	95.91
Andrews	60000	760154	151	$1.45202E+05$	07.89
apache2	715176	4817870	155	$6.65166E+06$	26.18
boneS01	127224	5516602	153	$1.13622E+06$	02.31
denormal	89400	1156224	153	$-2.88379E+05$	07.73
Dubcova3	146689	3636643	159	$-8.55189E+03$	46.57
ecology1	1000000	4996000	153	$3.61494E+06$	20.02
filter3D	106437	2707179	161	$-7.01011E+05$	79.95
finan512	74752	596992	151	$1.03471E+05$	19.67
G2_circuit	150102	726674	153	$6.58486E+05$	41.77
GaAsH6	61349	3381809	162	$2.32268E+05$	28.82
gas_sensor	66917	1703365	160	$-4.89303E+05$	90.37
H2O	67024	2216736	153	$3.08149E+05$	03.02
helm2d03	392257	2741935	153	$5.01026E+05$	14.31
Lin	256000	1766400	153	$1.60526E+06$	14.49
nasasrb	54870	2677324	161	$8.56473E+05$	62.37
offshore	259789	4242673	161	$4.84144E+06$	99.87
parabolic_fem	525825	3674625	153	$-4.83938E+05$	71.46
qa8fm	66127	1660579	153	$-5.51168E+05$	03.98
rail_79841	79841	553921	151	$-8.54968E+05$	15.09
s3dkq4m2	90449	4427725	161	$5.21115E+04$	73.77
shallow_water2	81920	327680	151	$1.95771E+06$	25.00
ship_003	121728	3777036	161	$3.05969E+06$	85.85
shipsec8	114919	3303553	164	$1.94819E+06$	82.96
t3dh_e	79171	4352105	156	$-1.28870E+06$	27.32
thermomech_TK	102158	711558	151	$4.85968E+05$	15.49
tmt_sym	726713	5080961	158	$1.00529E+06$	71.46
filter3D	106437	2707179	161	$-7.01011E+05$	79.95
G3_circuit	1585478	7660826	153	$6.72048E+06$	72.19
H2O	67024	2216736	153	$3.08149E+05$	03.02
SiO2	155331	11283503	153	$7.14208E+05$	17.34
thermal2	1228045	8580313	154	$1.63908E+06$	80.32

TABLE 4.5  
Newton iteration for sparse nonsymmetric matrices

Gallery	n	nnz	No. it.	Val(OAP)	Rem. En.(%)
af23560	23560	460598	2248	$8.74776E+04$	70.32
bayer04	20545	85537	183275	$-5.45190E+04$	80.21
bbmat	38744	1771722	2234	$4.73786E+04$	32.83
ec132	51993	380415	23389	$-2.73185E+05$	81.66
g7jac200sc	59310	717620	47245	$3.93891E+04$	86.40
gemat11	4929	33108	2780	$4.07095E+03$	84.70
graham1	9035	335472	4014	$-1.84675E+04$	51.59
hcircuit	105676	513072	34980	$-3.83585E+05$	88.59
hydr1	5308	22680	73772	$5.25311E+03$	78.65
jpwh_991	991	6027	151	$1.47688E+03$	16.44
lhr71c	70304	1528092	2227871	$-7.63013E+04$	83.56
mahindas	1258	7682	3485	$-6.49190E+01$	31.71
onetone1	36057	335552	23601	$1.13220E+05$	87.97
onetone2	36057	222596	24122	$1.13220E+05$	85.64
orani678	2529	90158	5073	$-1.57076E+02$	05.20
sherman3	5005	20033	168	$-2.62102E+04$	85.63
sherman5	3312	20793	1696	$6.67064E+03$	29.55

*Proof.* We only prove the last one since others are straightforward.

$$\begin{aligned}
Z_m &= \mathcal{R}(A^{(p_{m+1})} \text{diag}(V_m)) \\
&= \mathcal{R}(A^{(p_m)} \text{diag}(V_m) \circ A^{(p_{m+1}-p_m)}) \\
&= \mathcal{R}((\text{diag}(U_m) A^{(p_m)} \text{diag}(V_m)) \circ A^{(p_{m+1}-p_m)}) \\
&= \mathcal{R}(W_m \circ A^{(p_{m+1}-p_m)})
\end{aligned}$$

□

According to the previous proposition, we define the following iteration which we refer to as *deformed Sinkhorn iteration*.

$$\begin{aligned} W_0 &= \mathcal{C}(A^{(p_0)}); \\ W_m &= \mathcal{C}(Z_{m-1}), \quad c_m = (Z_{m-1}^T) \mathbb{1} \end{aligned} \quad (5.2)$$

$$Z_m = \mathcal{R}(W_m \circ A^{(p_{m+1}-p_m)}), \quad r_m = (W_m \circ A^{(p_{m+1}-p_m)}) \mathbb{1} \quad (5.3)$$

Here,  $r_m, c_m$  respectively are the vectors of row sums and column sums.

**5.2. Convergence to optimal assignment.** For an input matrix,  $A = (a_{ij})$ , assume that the deformed Sinkhorn iteration converges to a bistochastic matrix. Define the weight of a permutation,  $\sigma$ , with respect to  $A$ , to be  $\omega_\sigma(A) = \prod_i a_{i\sigma(i)}$ . If  $A$  has a support, it should have at least one optimal permutation as  $\sigma_{opt}$  with nonzero weight. It is evident that  $\sigma_{opt}$  is the optimal permutation for all the matrices  $W_m$  and  $Z_m$  produced by each deformed Sinkhorn iteration. Observe that for all permutations  $\sigma$  and  $\pi$ , the ratio  $\frac{\omega_\sigma(A)}{\omega_\pi(A)}$  is invariant if we multiply the matrix  $A$  by diagonal matrices. So it follows from the Equation 5.1 that

$$\gamma_m = \frac{\omega_\sigma(Z^m)}{\omega_\pi(Z^m)} = \gamma_{m-1} \left( \frac{\omega_\sigma(A)}{\omega_\pi(A)} \right)^{p_{m+1}-p_m} = \left( \frac{\omega_\sigma(A)}{\omega_\pi(A)} \right)^{p_{m+1}}$$

Thus, for all non optimal permutations such as  $\sigma$ ,  $\frac{\omega_\sigma(Z^m)}{\omega_{\sigma_{opt}}(Z^m)}$  will converge to zero when  $p_m \rightarrow \infty$ . Since in each iteration the weight of optimal permutation,  $\omega_{\sigma_{opt}}(Z^m)$ , is bounded above by 1, the weight of all non optimal permutations will converge to zero which yields the following lemma.

**LEMMA 5.2.** *Assume that the deformed Sinkhorn iteration converges to a matrix,  $Z$ , produced by the deformed Sinkhorn iteration when  $p_m \rightarrow \infty$ . If the original matrix  $A$  has a support, then all the permutations of  $Z$  have zero weight, except the optimal permutations of the original matrix  $A$ .*

Due to the theorem of Birkhoff-von Neumann, a square bistochastic matrix in  $\mathbb{R}$  is a convex combination of permutation matrices. Hence, all the nonzero entries of a bistochastic matrix belong to a permutation with nonzero weight. This statement together with the previous lemma yield the following theorem.

**THEOREM 5.3.** *For a non-negative matrix  $A$  which has a support, as  $p_m \rightarrow \infty$ , if the deformed Sinkhorn iteration converges to a matrix  $X$ , then all the nonzero entries of  $X$  belong to an optimal permutation of the original matrix.*

**5.3. Convergence to bistochastic matrix for positive matrices.** Recall that the rate of convergence of the classical Sinkhorn iteration is bounded above by  $\kappa(A)^2$  where  $\kappa(A) = \frac{\theta(A)^{1/2}-1}{\theta(A)^{1/2}+1}$ . The following theorem presents the main result of this section:

**THEOREM 5.4.** *Let  $A$  be a positive matrix. If  $p_m = a \log(m+1)$  where  $0 < a \log \theta < 2$ , then the deformed Sinkhorn iteration will converge to a bistochastic matrix and subsequently to a solution of optimal assignment of the original matrix  $A$ .*

The proof relies on the next lemmas. For a matrix  $A$ ,  $\theta(A) = \theta(A^T)$ , and for two diagonally equivalent matrices such as  $A$  and  $B$ ,  $\theta(A) = \theta(B)$ .

**LEMMA 5.5.** *For positive matrices  $A$  and  $B$ , diagonal matrix  $D$ , and  $d(x, x')$  the Hilbert projective metric, the following properties hold.*

1.  $d(Ax, Ax') \leq \kappa(A)d(x, x')$
2.  $d((A \circ B)x, x') \leq \log \frac{\max(B)}{\min(B)} + d(Ax, x')$

$$3. \kappa(AD \circ B) = \kappa(A \circ BD) = \kappa((A \circ B)D) = \kappa(D(A \circ B)) = \kappa(A \circ B)$$

*Proof.* The proof is straightforward.  $\square$

COROLLARY 5.6.  $\kappa(A)$  is invariant under  $\mathcal{R}$  or  $\mathcal{C}$  operators.

LEMMA 5.7. Let  $W_m$  and  $Z_m$  be the matrices in Equations (5.2,5.3) at iteration  $m$ . The following properties hold.

1.  $\kappa(Z_m) = \kappa(A^{(p_{m+1})})$
2.  $\kappa(W_m) = \kappa(A^{(p_m)})$

*Proof.* The proof is straightforward by using the induction on  $m$ .  $\square$

The next lemma is similar to Lemma 2 in [FL89], where the classical Sinkhorn iteration is considered.

LEMMA 5.8. Let  $r_m, c_m$  be the vectors defined in Equation (5.2,5.3) at iteration  $m$  and  $M = \frac{\max(A)}{\min(A)}$  then,

$$\begin{aligned} d(r_m, \mathbb{1}) &\leq (p_{m+1} - p_m) \log M + (p_m - p_{m-1}) \kappa(A^{(p_m)}) \log M \\ &\quad + \kappa(A^{(p_m)}) \kappa(A^{(p_{m-1})}) d(r_{m-1}, \mathbb{1}) \\ d(c_m, \mathbb{1}) &\leq (p_m - p_{m-1}) \log M + (p_m - p_{m-1}) \kappa(A^{(p_{m-1})}) \log M \\ &\quad + \kappa^2(A^{(p_{m-1})}) d(c_{m-1}, \mathbb{1}) \end{aligned}$$

*Proof.* Let  $\mathbb{1}/V$  indicates the entrywise inverse of a given vector,  $V$ . We have,

$$\begin{aligned} r_m &= (W_m \circ A^{(p_{m+1}-p_m)}) \mathbb{1} = (Z_{m-1} \text{diag}(\mathbb{1}/c_m) \circ A^{(p_{m+1}-p_m)}) \mathbb{1} \\ &= (Z_{m-1} \circ A^{(p_{m+1}-p_m)}) \text{diag}(\mathbb{1}/c_m) \mathbb{1} = (Z_{m-1} \circ A^{(p_{m+1}-p_m)})(\mathbb{1}/c_m), \end{aligned}$$

so

$$\begin{aligned} d(r_m, \mathbb{1}) &= d((Z_{m-1} \circ A^{(p_{m+1}-p_m)})(\mathbb{1}/c_m), Z_{m-1} \mathbb{1}) \\ &\leq (p_{m+1} - p_m) \log M + \kappa(Z_{m-1}) d(c_m, \mathbb{1}) \\ &= (p_{m+1} - p_m) \log M + \kappa(A^{(p_m)}) d(c_m, \mathbb{1}). \end{aligned}$$

Also

$$\begin{aligned} d(c_m, \mathbb{1}) &= d((W_{m-1}^T \circ A^{(p_m-p_{m-1})^T})(\mathbb{1}/r_{m-1}), W_{m-1}^T \mathbb{1}) \\ &\leq (p_m - p_{m-1}) \log M + \kappa(W_{m-1}^T) d(\mathbb{1}/r_{m-1}, \mathbb{1}) \\ &= (p_m - p_{m-1}) \log M + \kappa(W_{m-1}) d(r_{m-1}, \mathbb{1}) \\ &= (p_m - p_{m-1}) \log M + \kappa(A^{(p_{m-1})}) d(r_{m-1}, \mathbb{1}), \end{aligned}$$

then

$$\begin{aligned} d(r_m, \mathbb{1}) &\leq (p_{m+1} - p_m) \log M + (p_m - p_{m-1}) \kappa(A^{(p_m)}) \log M \\ &\quad + \kappa(A^{(p_m)}) \kappa(A^{(p_{m-1})}) d(r_{m-1}, \mathbb{1}) \end{aligned}$$

The second statement is established in a similar way.  $\square$

LEMMA 5.9. Assume that  $p_m = a \log(m+1)$ , where  $0 < a \log \theta(A) < 2$ . Then we have  $\lim_{m \rightarrow \infty} d(c_m, \mathbb{1}) = 0$ .

*Proof.* Since

$$\begin{aligned} d(c_m, \mathbb{1}) &= a \log \frac{m+1}{m} \log M + a \log \frac{m+1}{m} \kappa(A^{(p_{m-1})}) \log M \\ &\quad + \kappa^2(A^{(p_{m-1})}) d(c_{m-1}, \mathbb{1}) \\ &< \frac{2a \log M}{m} + \kappa^2(A^{(p_{m-1})}) d(c_{m-1}, \mathbb{1}) . \end{aligned}$$

Let  $\beta_1 := d(c_1, \mathbb{1})$ , and define the sequence  $\beta_m$  by  $\beta_m := f_{m-1}(\beta_{m-1})$ , where

$$f_{m-1}(x) = \frac{2a \log M}{m} + \kappa^2(A^{(p_{m-1})})x .$$

Since every function  $f_m$  is nondecreasing, an immediate induction shows that  $d(c_m, \mathbb{1}) \leq \beta_m$ , for all  $m \geq 1$ , and so, it suffices to show that  $\lim_m \beta_m = 0$ .

Let  $l_m$  be the fixed point of  $f_{m-1}$ . Setting

$$\alpha := \frac{a \log \theta(A)}{2}$$

and observing that

$$1 - \kappa^2(A^{(p_{m-1})}) = \frac{4m^{-\alpha}}{(1 + m^{-\alpha})^2} ,$$

we get

$$l_m = \frac{2a \log M}{m(1 - \kappa^2(A^{(p_{m-1})}))} = \frac{a \log M}{2} \frac{(1 + m^{-\alpha})^2}{m^{1-\alpha}} .$$

Since  $0 < \alpha < 1$ , one readily checks that the sequence  $l_m$  decreases with  $m$  and converges to zero. If  $\beta_{m+1} \leq l_m$  for every  $m$ , then  $\lim_{m \rightarrow \infty} \beta_m \leq \lim_{m \rightarrow \infty} l_m = 0$ , and the result is established. Assume now that  $\beta_{m+1} > l_m$  for some  $m$ . Define  $\delta_k := \beta_{k+1} - l_k$  for all  $k \geq m$ . Observe that

$$\delta_{k+1} = f_k(\beta_k) - f_k(l_k) = \kappa^2(A^{(p_k)})(\beta_k - l_k) = \kappa^2(A^{(p_k)})\delta_k + \kappa^2(A^{(p_k)})(l_{k-1} - l_k) .$$

Using the fact that  $\kappa^2(A^{(p_r)}) \leq 1$  holds for all  $r$ , an immediate induction yields

$$\delta_k \leq \left( \prod_{r=m}^{k-1} \kappa^2(A^{(p_r)}) \right) \delta_m + l_m - l_k, \quad \forall k \geq m+1 . \quad (5.4)$$

Since  $1 - \kappa^2(A^{(p_r)}) \sim 4r^{-\alpha}$ , we have

$$\prod_{r=m}^{\infty} \kappa(A^{(p_r)}) = 0$$

Letting  $k \rightarrow \infty$  in (5.4), we get  $\limsup_{k \rightarrow \infty} \delta_k \leq l_m$ . Since this holds for all  $m$ , it follows that  $\limsup_{k \rightarrow \infty} \delta_k \leq 0$ , and so  $\limsup_{k \rightarrow \infty} \beta_{k+1} = \limsup_{k \rightarrow \infty} \delta_k + l_k \leq \limsup_{k \rightarrow \infty} \delta_k + \lim_{k \rightarrow \infty} l_k = 0$ . Hence,  $\beta_k$  converges to zero.  $\square$

The proof of the Theorem 5.4 is achieved since  $\lim_{m \rightarrow \infty} d(c_m, \mathbb{1}) = 0$  yields  $\lim_{m \rightarrow \infty} d(r_m, \mathbb{1}) = 0$

**6. Conclusion.** We considered the connection between the entropy maximization problem and the optimal assignment problem. This allows us to propose an algorithm which can be used as a preprocessing in the solution of large scale optimal assignment problems to reduce the size of the input problem in terms of memory requirements.

Two variants of the algorithm have been implemented. The first variant, which is based on Sinkhorn iteration, shows a generally reasonable convergence for dense matrices, with a reduction of up to 99% of the input problem. However the algorithm

works slowly for sparse matrices. This version of the algorithm can be efficiently used as a parallel preprocessing to reduce the size of the input problem in very large dense optimal assignment problems.

Another variant of the algorithm, implemented by using the Newton iteration, shows fast convergence for all dense matrices and sparse symmetric matrices. However the convergence speed for sparse nonsymmetric matrices is slow.

The last part of the paper concerns a new iterative method that we refer to as *deformed-Sinkhorn iteration*. It is proved that the iteration converges to the solution of optimal assignment problem, if the input matrix is positive and if it has only one optimal permutation. For positive matrices with more than one optimal permutation, the iteration converges to a matrix for which all the nonzero entries belong to at least one optimal permutation.

*Acknowledgment.* The authors thank Jean-Charles Gilbert for his comments on an early version of this manuscript.

## REFERENCES

- [ADRU08] P. Amestoy, I. S. Duff, D. Ruiz, and B. Uçar. A parallel matrix scaling algorithm. In *High Performance Computing for Computational Science - VECPAR 2008*, volume 5336 of *Lecture Notes in Computer Science*, pages 301–313. Springer Berlin / Heidelberg, 2008.
- [BDM09] R. Burkard, M. Dell’Amico, and S. Martello. *Assignment problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2009.
- [BFH<sup>+</sup>03] Y. Brenier, U. Frisch, M. Henon, G. Loeper, S. Matarrese, R. Mohayaee, and A. Sobolevskii. Reconstruction of the early universe as a convex optimization problem. *Mon.Not.Roy.Astron.Soc.*, 346:501–524, 2003.
- [Bir46] G. Birkhoff. Three observations on linear algebra. *Univ. Nac. Tucumán. Revista A.*, 5:147–151, 1946.
- [BK76] T. Bewley and E. Kohlberg. The asymptotic theory of stochastic games. *Math. Oper. Res.*, 1(3):197–208, 1976.
- [BLN94] J. M. Borwein, A. S. Lewis, and R. D. Nussbaum. Entropy minimization, DAD problems, and doubly stochastic kernels. *J. Funct. Anal.*, 123(2):264–307, 1994.
- [BPR06] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in real algebraic geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, second edition, 2006.
- [BR97] R. B. Bapat and T. E. S. Raghavan. *Nonnegative matrices and applications*, volume 64 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1997.
- [Bru74] R. A. Brualdi. The DAD theorem for arbitrary row sums. *Proc. Amer. Math. Soc.*, 45:189–194, 1974.
- [BT09] L. Buš and P. Tvrdík. Towards auction algorithms for large dense assignment problems. *Comput. Optim. Appl.*, 43(3):411–436, 2009.
- [CWC<sup>+</sup>96] Y.-Q. Cheng, V. Wu, R. T. Collins, A. R. Hanson, and E. M. Riseman. Maximum-weight bipartite matching technique and its application in image feature matching. In *In Proc. SPIE Visual Comm. and Image Processing*, 1996.
- [DK69] E. A. Dinic and M. A. Kronrod. An algorithm for solving the assignment problem. *Dokl. Akad. Nauk SSSR*, 189:23–25, 1969.
- [DK00] I. S. Duff and J. Koster. On algorithms for permuting large entries to the diagonal of a sparse matrix. *SIAM J. Matrix Anal. Appl.*, 22(4):973–996, 2000.
- [DRU08] I. S. Duff, D. Ruiz, and B. Uçar. Computing a class of bipartite matchings in parallel. Presentation at SIAM 13th Conference on Parallel Processing for Scientific Computing (PP08), Atlanta, GA, USA, March 2008.
- [EK70] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. In *Combinatorial Structures and their Applications (Proc. Calgary Internat. Conf., Calgary, Alta., 1969)*, pages 93–96. Gordon and Breach, New York, 1970.
- [FL89] J. Franklin and J. Lorenz. On the scaling of multidimensional matrices. *Linear Algebra Appl.*, 114/115:717–735, 1989.

- [FRt97] S.-C. Fang, J. R. Rajasekera, and H.-S. J. Tsao. *Entropy optimization and mathematical programming*. International Series in Operations Research & Management Science, 8. Kluwer Academic Publishers, Boston, MA, 1997.
- [FT87] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. Assoc. Comput. Mach.*, 34(3):596–615, 1987.
- [HCLH90] C.-Y. Huang, Y.-S. Chen, Y.-L. Lin, and Y.-C. Hsu. Data path allocation based on bipartite weighted matching. *Design Automation Conference*, pages 499–504, 1990.
- [Hol93] L. Holm. Protein Structure Comparison by Alignment of Distance Matrices. *Journal of Molecular Biology*, 233(1):123–138, September 1993.
- [KK92] L. Khachiyan and B. Kalantari. Diagonal matrix scaling and linear programming. *SIAM J. Optim.*, 2(4):668–672, 1992.
- [Kni08] P. A. Knight. The Sinkhorn-Knopp algorithm: convergence and applications. *SIAM J. Matrix Anal. Appl.*, 30(1):261–275, 2008.
- [KR07] P. A. Knight and D. Ruiz. A fast algorithm for matrix balancing. In Andreas Frommer, Michael W. Mahoney, and Daniel B. Szyld, editors, *Web Information Retrieval and Linear Algebra Algorithms*, number 07071 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2007. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.
- [Kuh55] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Res. Logist. Quart.*, 2:83–97, 1955.
- [LCL04] Lin, Y. H., Chang, H. C., and Lin, Y. L. A Study on Tools and Algorithms for 3-D Protein Structures Alignment and Comparison. In *Int. Computer Symposium*, pages 1000–1005, December 2004.
- [LD03] X. S. Li and J. W. Demmel. SuperLU\_DIST: A Scalable Distributed-memory Sparse Direct Solver for Unsymmetric linear systems. *ACM Transactions on Mathematical Software*, 29(2), 2003.
- [LO94] Y. Lee and J. B. Orlin. On very large scale assignment problems. In *Large scale optimization (Gainesville, FL, 1993)*, pages 206–244. Kluwer Acad. Publ., Dordrecht, 1994.
- [LSW00] N. Linial, A. Samorodnitsky, and A. Wigderson. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. *Combinatorica*, 20(4):545–568, 2000.
- [Mar] T. Markwig. A field of generalised puiseux series for tropical geometry. to appear in *Rend. Semin. Mat. Torino* (2009).
- [Mir60] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *Quart. J. Math. Oxford Ser. (2)*, 11:50–59, 1960.
- [MPV] M. Mezard, G. Parisi, and M. Virasoro. *Spin Glass Theory and Beyond (World Scientific Lecture Notes in Physics, Vol 9)*. World Scientific Publishing Company.
- [MS69] M. V. Menon and H. Schneider. The spectrum of a nonlinear operator associated with a matrix. *Linear Algebra and Appl.*, 2:321–334, 1969.
- [MZ01] G. Malajovich and J. P. Zubelli. Tangent Graeffe iteration. *Numer. Math.*, 89(4):749–782, 2001.
- [NN94] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13 of *SIAM Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.
- [ON96] M. Olschowska and A. Neumaier. A new pivoting strategy for Gaussian elimination. *Linear Algebra Appl.*, 240:131–151, 1996.
- [Par02] G. Parisi. Euclidean random matrices, the glass transition and the boson peak. *The European Physical Journal E: Soft Matter and Biological Physics*, 9(3):213–218, November 2002.
- [Sch89] M. H. Schneider. Matrix scaling, entropy minimization, and conjugate duality. I. Existence conditions. *Linear Algebra Appl.*, 114/115:785–813, 1989.
- [SK67] R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific J. Math.*, 21:343–348, 1967.
- [SZ90] M. H. Schneider and S. A. Zenios. A comparative study of algorithms for matrix balancing. *Oper. Res.*, 38:439–455, May 1990.
- [Tar51] A. Tarski. *A decision method for elementary algebra and geometry*. University of California Press, Berkeley and Los Angeles, Calif., 1951. 2nd ed.
- [Vir01] O. Viro. Dequantization of real algebraic geometry on logarithmic paper. In *European Congress of Mathematics, Vol. I (Barcelona, 2000)*, volume 201 of *Progr. Math.*, pages 135–146. Birkhäuser, Basel, 2001.